

Claims

1. (Currently amended) A method of reading stored data from an array of independent disk drives so as to provide synchronous data transfer into a buffer, the method comprising:

for each disk drive in the array, providing a corresponding two-port memory for receiving and storing read data responsive to timing signals provided by the respective drive;

initiating a READ command to each of the drives of the array, thereby causing each of the drives to retrieve selected elements of its stored data, and to transfer the retrieved data from the drive into ~~the~~ its corresponding two-port memory using the timing signals provided by the respective drive;

monitoring each of the two-port memories to detect a non-empty condition[[,]]
~~implying receipt of transferred data in the memory from the corresponding disk drive;~~

waiting until all of the two-port memories indicate such a non-empty condition;

then synchronously reading the ~~stored~~ transferred data from all of the two-port memories, thereby forming synchronous read data, and writing the synchronous read data into the buffer; and

repeating said monitoring, waiting; reading and writing into the buffer steps until completion of a read operation initiated by the said READ command.

2. (Currently amended) A method of reading stored data according to claim 1 wherein:

the ~~stored~~ transferred data includes user data as well as redundant data sufficient to enable reconstruction of all of the user data in the event of a failure of any single drive of the array and the method further comprising,

in the event that one of the disk drives fails, executing said initiating, monitoring, waiting and synchronously reading steps only with respect to the non-failed drives; and

dynamically regenerating missing data corresponding to the failed drive “~~on the fly~~” from the synchronous read data.

3. (Currently amended) A method of reading stored data from an array according to claim 1 wherein each two-port memory comprises a FIFO first-in first-out (FIFO) memory.

4. (Currently amended) A method of reading stored data from an array according to claim 3 wherein the array comprises a redundant array.

5. (Currently amended) A method of reading stored data from an array according to claim 4 and further comprising ~~regenerating data “on the fly”~~ in the event that one of the disk drives of the redundant array has failed , dynamically regenerating missing data corresponding to the failed drive from the synchronous read data.

6. (Currently amended) A method of reading stored data from an array according to claim 1 wherein at least one of the disk drives is coupled to its corresponding two-port memory via ~~the read operation is effected via a UDMA~~ an Ultra-DMA (UDMA) interface to at least one of the disk drives.

7. (Currently amended) A method of reading data from an array according to claim 1 wherein the read operation is effected via a corresponding UDMA Ultra-DMA (UDMA) interface to each of the disk drives.

8. (Currently amended) A method of reading data from an array according to claim 1 wherein said synchronously reading the stored data from all of the two-port memories comprises asserting a common read enable signal to each of the memories.

9. (Currently amended) A method of reading data from an array according to claim 1 wherein said synchronously reading the stored data from all of the two-port memories is conducted over a single DMA direct memory access (DMA) channel.

10. (Currently amended) A method of reading stored data from a redundant array of independent disk drives (RAID array) comprising:

for each disk drive in the ~~redundant~~ RAID array, providing a corresponding FIFO first-in first-out (FIFO) memory arranged for receiving and storing **read data** using timing signals provided by the respective drive;

initiating a READ command to each of the drives of the RAID array, thereby causing each of the drives to retrieve selected elements of its stored data, and to **transfer the retrieved data** from the drive into ~~the~~ its corresponding FIFO memory using the timing signals provided by the respective drive;

monitoring each of the FIFO memories to detect a non-empty condition[[,]]
~~implying receipt of data in the FIFO memory from the corresponding disk drive;~~
waiting until all of the FIFO memories indicate such a non-empty condition;
then synchronously reading the ~~stored~~ transferred data from all of the FIFO
memories, thereby forming synchronous read data;
writing the synchronous read data into a common buffer; and
repeating said monitoring, waiting, reading and writing steps until completion of a
read operation initiated by the READ command.

11. (Currently amended) A method of reading stored data according to claim 10 wherein the stored data is word striped over the redundant array.

12. (Currently amended) A method of reading stored data according to claim 10 and further comprising, in the event that one of the disk drives fails to ~~provide~~ transfer read data to its associated FIFO memory, regenerating the missing read data “on the fly” from the synchronous read data.

13. (Currently amended) A method of reading data according to claim 10 wherein each of the drives is coupled to its associated FIFO memory via a ~~UDMA~~ Ultra-DMA (UDMA) interface.

14. (Original) A method of reading data according to claim 10 wherein the synchronous transfer of read data into the common buffer is implemented with a single address counter and a common FIFO read enable signal.

15. (Original) A method of reading data from an array according to claim 10 wherein each synchronous transfer of read data into the common buffer stores 64-bits of read data.

16. (Original) A method of reading data from an array according to claim 10 and further comprising providing a FIFO memory in the data path between the individual drive FIFO memories and the common buffer.

17. (Currently amended) An improved ~~RAID~~ redundant array of independent disk drives (RAID) array disk array controller comprising:

a plurality of disk drive interfaces, each interface for attaching at least one physical disk ~~drives~~ drive;

a corresponding two-port memory associated with each one of the disk drive interfaces, each two-port memory arranged to store read data provided by ~~the~~ its associated disk drive in a disk read operation and, conversely, to provide write data that was previously-stored in the two-port memory to ~~the~~ its associated disk drive in a disk write operation;

a logic circuit coupled to all of the two-port memories for detecting when all of the two-port memories have data stored therein for a read operation or available space therein for a write operation;

control circuitry responsive to the logic circuit for synchronously reading data from all of the two-port memories only when all of the two-port memories have data stored therein, thereby forming synchronous read data;

the control circuitry further responsive to the logic circuit for detecting that all of the two-port memories have space therein and synchronously writing data to all of the two port memories thereby forming synchronous write data;

first redundant data circuitry for regenerating missing data “on the fly” from the synchronous read data in the event that one of the disk drives fails to provide read data to its associated two-port memory in a read operation; and

second redundant data circuitry for generating redundant data “on the fly” from the synchronous write data for storing in the array[[]]

and wherein:

one of the disk drive interfaces is designated for connection to a redundant disk drive for storing redundant data so that the synchronous read data includes redundant data;

the first redundant data circuitry for regenerating missing data is coupled to all of the two-port memories to receive the synchronous read data, including the redundant data, and further is arranged to compute a redundant data operation across the synchronous read data so as to provide corrected data in the event of a failed drive without delaying output of the read data.

18. (Original) An improved RAID disk array controller according to claim 17 and wherein each two-port memory has multiplexers for exchanging its input and output ports depending on the data transfer direction.

19. (Currently amended) An improved RAID disk array controller according to claim 17 wherein each two-port memory comprises a ~~FIFO~~ first-in first-out (FIFO) memory.

20. (Currently amended) An improved RAID disk array controller according to claim 17 wherein the common buffer comprises ~~DRAM~~ dynamic random-access memory (DRAM).

21. (Currently amended) An improved RAID disk array controller according to claim 17 and further comprising a single address counter arranged for addressing the buffer for transfers between the buffer and the ~~FIFO~~ first-in first-out (FIFO) memories in either direction.

22. (Currently amended) An improved disk array controller according to claim 17 wherein at least one disk drive interface implements a ~~ATA/ATAPI~~ AT Attachment and ATA Packet Interface (ATA/ATAPI) protocol.

23. (Currently amended) An improved disk array controller according to claim 17 wherein all of the disk drive interfaces implement a ~~ATA/ATAPI~~ AT Attachment and ATA Packet Interface (ATA/ATAPI) protocol.

24. (Original) An improved disk array controller according to claim 17, implemented on a motherboard.

25. (Original) An improved disk array controller according to claim 17, implemented on a Host Bus Adapter.

26. (Currently amended) A method of writing data into an array of independent disk drives, the method comprising:

- providing a buffer for storing write data;
- for each disk drive in the array, providing a corresponding two-port memory for receiving and storing write data, ~~the two-port memory~~;
- monitoring each of the two-port memories to detect a non-full condition;
- waiting until all of the two-port memories indicate such a non-full condition;
- then reading write data from the buffer;

computing redundant data from said write data;
synchronously storing the write data and the computed redundant data into the two-port memories via a first port of each two-port memory; and
~~substantially concurrently;~~ while synchronously storing the write data and the computed redundant data into the two-port memories, transferring previously-stored data from a second port of each of the two-port memories into the its corresponding disk ~~drives~~ drive, in each case transferring the previously-stored data responsive to timing control provided by the ~~respective~~ corresponding disk drive.

27. (Currently amended) A method of ~~storing~~ writing data into an array according to claim 26 and further comprising stalling said storing step whenever any of the two-port memories ~~becomes~~ become full, but only with regard to the full memory, while allowing said synchronously storing the write data to continue into the non-full two-port memories.

28. (Currently amended) A method of ~~storing~~ writing data into an array according to claim 27 wherein each two-port memory comprises a ~~FIFO~~ first-in first-out (FIFO) memory.

29. (Currently amended) A method of ~~storing~~ writing data into an array according to claim 28 wherein the ~~write operation is effected via~~ array includes a UDMA interface to at least one of the disk drives.

30. (Currently amended) A method of storing data into an array according to claim 28 wherein the write operation is effected via a corresponding UDMA interface to each of the disk drives.

31. (Currently amended) A method of storing data into an array according to claim ~~[[27]]~~ 28 wherein said synchronously storing the write data ~~into the FIFOs~~ comprises asserting a common write strobe coupled to all of the FIFO memories.